Ontology-based semantic annotation: an automatic hybrid rule-based method

Sondes Bannour, Laurent Audibert, Henry Soldano

Ontology-based semantic annotation: an automatic hybrid rule-based method

Sondes Bannour, Laurent Audibert and Henry Soldano
LIPN, UMR 7030 CNRS
Université Paris 13, Sorbonne Paris Cité, F-93430, Villetaneuse, France
firstname.lastname@lipn.univ-paris13.fr

Abstract

In the perspective of annotating a text with respect to an ontology, we have participated in the subtask 1 of the BB BioNLP-ST whose aim is to detect, in the text, Bacteria Habitats and associate to them one or several categories from the OntoBiotope ontology provided for the task. We have used a rule-based machine learning algorithm (WHISK) combined with a rule-based automatic ontology projection method and a rote learning technique. The combination of these three sources of rules leads to good results with a SER measure close to the winner and a best F-measure.

1 Introduction

Ontology-based semantic annotation consists in linking fragments of a text to elements of a domain ontology enabling the interpretation and the automatic exploitation of the texts content. Many systems annotate texts with respect to an ontology (Dill et al., 2003). Some of them use machine-learning techniques to automate the annotation process (Ciravegna, 2000).

On one side, machine-learning techniques depend strongly on the amount and quality of provided training data sets and do not use information available in the ontology. On the other side, using the ontology to project its elements onto the text depends strongly on the richness of the ontology and may neglect important information available in texts.

Our participation in the subtask 1 (entity detection and categorization) of the BB BioNLP-ST leverages the provided OntoBiotope ontology and the training and development data sets preprocessed using our annotation platform based on UIMA (Ferrucci and Lally, 2004) (section 2). We first tested, on the development set, a rule-based machine-learning algorithm (WHISK (Soderland et al., 1999)) that used training set examples (section 3). Its results are limited because of the weaknesses of training data (section 4). We, then, computed a rule-based automatic ontology projection method consisting in retrieving from the text field information content provided by the ontology (*eg.* name of the concept). Thanks to the wealth of the OntoBiotope ontology, this method gave good results (section 5) that have been improved by adding a rote learning technique that uses training examples and some filtering techniques (section 6). Finally, we combined our method with WHISK results, which slightly improved the F-measure (section 7) on the development data.

2 TextMarker and data preprocessing

In a rule-based information extraction or semantic annotation system, annotation rules are usually written by a domain expert. However, these rules can be learned using a rule-based learning algorithm. The TextRuler system (Kluegl et al., 2009) is a framework for semi-automatic development of rule-based information extraction applications that contains some implementations of such algorithms ($(LP)^2$ (Ciravegna, 2001; Ciravegna, 2003), WHISK (Soderland et al., 1999), RAPIER (Califf and Mooney, 2003), BWI (Freitag and Kushmerick, 2000) and WIEN (Kushmerick et al., 1997)). TextRuler is based on Apache UIMA TextMarker which is a rule-based script language.

TextMarker is roughly similar to JAPE (Cunningham et al., 2000), but based on UIMA (Ferrucci and Lally, 2004) rather than GATE (Cunningham, 2002). According to some users experiences, it is even more complete than JAPE. Here is an example that gives an idea about how to write and use TextMarker rules: Given an UIMA type system that contains the types SPACE (whitespace) and Lemma (with a feature "lemma" containing the lemmatized form of the matched

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 139–143,
Sofia, Bulgaria, August 9 2013.

word), the following rule can be used to recognize the term "human body" in whatever form it appears in the text (singular, plural, uppercase, lowercase):

```
Lemma{FEATURE("lemma","human")}
  SPACE Lemma{FEATURE("lemma","body")
  --> MARK(Habitat, 1, 2, 3)};
```

This rule allows the creation of an annotation called "Habitat" that covers the three matched patterns of the condition part of the rule.

To be able to use TextMarker, we have used our annotation platform based on UIMA to preprocess data with:

- Tokenisation, lemmatisation, sentence splitting and PoS-tagging of input data using BioC (Smith et al., 2004; Liu et al., 2012).
- Term extraction using BioYatea (Golik et al., 2013), a term extractor adapted to the biomedical domain.
- Bacteria Habitat annotation to train learning algorithms using annotation files provided in this task (.a2).

For simplicity reasons, we do not take into account discontinuous annotations. We consider a discontinuous annotation as the smallest segment that include all fragments.

3 Rule Learning using WHISK

"In the subtask 1 of the BB BioNLP-ST, participants must detect the boundaries of Bacteria Habitat entities and, for each entity, assign one or several concepts of the OntoBiotope ontology." Should we decompose the task into two subtasks like it is suggested in the task formulation : (1) entity detection and (2) categorization ? To answer this question, we have conducted two experiments.

- Learning the root concept Habitat without assigning a Category to matched terms.
- Learning Bacteria Categories directly: each Habitat Category is learned independently.

For the two experiments we considered only Categories that have more than two examples in the training set to train WHISK. Results are shown in Table 1:

Experiment	Precision	Recall	F-measure
Habitats learning	76.9%	24.5%	37.2%
Categories learning	77.3%	24%	36.6%

Table 1: Habitats learning *vs* Categories learning

WHISK gives an acceptable precision but a low recall (the explanation is provided in section 4) for both experiments. There is no big difference between the two experiments' results: WHISK doesn't generalize over Habitats Categories. Learning Habitat Categories seems to be the easier and safer way to use WHISK in this task.

4 Weaknesses of training examples explain poor rule learning results

	Training	Development	Total
Nb. Concepts:	333	274	**491**
Nb. Habitat:	934	611	1545
Nb. Annotation:	948	626	1574
Nb. C. 1 Instance:	182	179	272
Nb. C. 2 Instances:	66	41	86
Nb. C. $>$ 2 Instances:	27	15	**133**
Number of concepts in ontology:			**1756**

Table 2: Figures on provided data

A close look at data samples helps understand why the WHISK algorithm did not obtain good results. Table 2 exhibits some figures on training and development data:

- 158 of the 274 concepts (58%) present in the development data do not appear in the training data.
- Concepts present in sample data account for 19% of the ontology for the training data, 16% for the development data and 28% for their combination.
- Obviously, it is difficult for a machine learning algorithm to learn (*i.e.* generalize) on only one instance. This is the case for 55% (272) of the concepts considering both the training and the development sample data.
- If we consider that at least 3 instances are needed to apply a machine learning algorithm, only 27% of concepts present in the training or development data are concerned. This means that the ontology coverage is less than 8%.

The conclusion is that training data are too small to lead to a high performance recall for a machine learning algorithm based exclusively on these data.

5 The wealth of the ontology helps build an efficient ontology-based rule set

The BB BioNLP-ST's subtask 1 provides the OntoBiotope ontology used to tag samples. For ex-

ample, the information provided by the ontology for the concept MBTO:00001516 is

```
[Term]
id: MBTO:00001516
name: microorganism
exact_synonym: "microbe" [TyDI:23602]
related_synonym: "microbial" [TyDI:23603]
is_a: MBTO:00000297 ! living organism
```

Text segments tagged with this concept in examples are : microbe, microbial, microbes, microorganisms, harmless stomach bugs.

One can notice that the name, exact_synonym and related_synonym field information provided by the ontology can help identify these segments. If this strategy works, it will be a very robust one because it is not sample dependent and it is applicable for all the 1756 concepts present in the ontology.

The main idea is to directly search and tag in the corpus the information provided by the content of fields name, exact_synonym and related_synonym of the ontology. Of course, projecting them directly on samples raises inflection issues. Our corpus provides two levels of lemmatisation to avoid inflection problems: one from BioC and the other from BioYaTeA. Our experiments show that using the two of them in conjunction with the token level (without any normalisation of words) provides the best results. For example, the rules to project name field of MBTO:00001516 are:

```
Token{REGEXP("^microorganism$")
  -> MARKONCE(MBTO:00001516,1)} ;
Lemma{FEATURE("lemma","microorganism$")
  -> MARKONCE(MBTO:00001516,1)} ;
Term{FEATURE("lemma","microorganism$")
  -> MARKONCE(MBTO:00001516,1)} ;
```

Table 3 provides results obtained on development data. We have also used training data to generate rote learning rules introduced in the next section.

Rule set name	Precision	Recall	F-measure
name:	67.4%	61.2%	64.2%
exact_synonym:	61.2%	4.2%	7.8%
related_synonym:	26.6%	5.9%	9.7%
rote learning:	63.6%	50.2%	56.1%
all together:	58.9%	73.8%	65.5%

Table 3: Performances of some sets of rules

6 Improving ontology-based rules

Rote learning rules

Results obtained for name and exact_synonym rules in Table 3 are very encouraging. We can apply the same strategy of automatic rule generation from training data to text segments covered by training examples. Projection rules are generated, as described in section 5, for each example segment using the associated concept's name as the rule conclusion. This is a kind of rote learning. Of course, we use an appropriate normalised version of example segment to produce appropriate rules based on BioC lemmatisation and BioYaTeA lemmatisation[1]. For example, rote learning rules for the segment harmless stomach bugs tagged as MBTO:00000297 in trainning data are:

```
Token{REGEXP("^harmless$")}
  Token{REGEXP("^stomach$")}
  Token{REGEXP("^bugs$")
  -> MARKONCE(MBTO:00001516,1,3)} ;
Lemma{FEATURE("lemma","harmless")}
  Lemma{FEATURE("lemma","stomach")}
  Lemma{FEATURE("lemma","bug")
  -> MARKONCE(MBTO00001516,1,3)} ;
```

Rule sets filtering

Rule set name	Precision	Recall	F-measure
name:	87.6%	55.1%	67.6%
exact_synonym:	94.4%	2.7%	5.3%
related_synonym:	71.4%	2.4%	4.6%
rote learning:	75.8%	44%	55.8%
all together:	80.9%	63.4%	71.1%
all together bis:	81.4%	63.4%	71.2%

Table 4: Performances of sets of filtered rules

A detailed analysis shows that our strategy works well on the majority of concepts, but produces poor results for some concepts. To overcome this limitation, we have adopted a strategy consisting in filtering (deleting) rules that produce lots of erroneous matches. More precisely, we have deleted rules that match at least one time and that conclude on a concept that obtains both a precision less or equal to 0.66 and a F-measure less or equal to 0.66. This filtering is computed on training data. Table 4 shows performances on development data obtained by filtered versions of rules of table 3.

Rule sets combination

Our goal is to maximise the F-measure. F-measure in table 4 for exact_synonym and related_synonym rules is worse than in table 3 because of the decrease of the recall. But the combination of the four simple rule sets allows to recover some of the lost recall. The significative im-

[1]The information from BioYaTeA exists only for segments identified as a term.

provement of precision finally leads to an overall improvement of the F-measure (*all together* in table 4). Removing either one of the four sets of rules that constitute the *all together* set of rules from table 4 leads systematically to a decrease of the F-measure.

Embedded rules removing

We have noticed a phenomenon that decreases precision and that can be corrected when combining ontology-based sets of rules with the *rote learning* set of rules. To illustrate it, the name of the concept `MBTO:00002027` is `plant`. Among examples tagged with this concept, we can find `healthy plants`. The *name* rule set matches on `plants` and tags it with `MBTO:00002027` (which is a mistake), while the *rote learning* rule set matches on `healthy plants` and tags it with `MBTO:00002027`. It is possible to correct this problem by a simple rule that unmarks such embedded rules:

```
MBTO:00002027{ PARTOFNEQ( MBTO:00002027 )
  -> UNMARK( MBTO:00002027 ) } ;
```

We have generated such a rule systematically for all the concepts of the ontology to remove a few mistakes (*all together bis* set of rules in table 4).

7 Adding Learned rules

Finally, we have completed the *all together bis* set of filtered rules with the rules produced by the WHISK algorithm. The difference between *all together bis* + *whisk* set of rules and the *submitted* set of rules is that, by mistake, the last one did not contain the *related_synonym* rule set.

It is important to mention that all rules may apply simultaneously. There is also no execution order between them except for rules that remove embedded ones which must be applied at the end of the rules set but before WHISK rules.

Rule set name	Precision	Recall	F-measure
all together bis:	81.4%	63.4%	71.2%
all[...] + whisk:	79.1%	65%	71.4%
submitted:	79.3%	64.4%	71.1%

Table 5: Performances of final sets of rules on dev data

Table 5 summarises performances achieved by our final rule sets. *Precision*, *Recall* and *F-measure* are computed on the development data with rules based on the training data.

Table 6 summarises performances on test data with the evaluator's measures achieved by our final rule sets based on training plus development data.

Rule set name	Precision	Recall	F1	SER
all together bis:	66.5%	61.4%	63.9%	42.5%
all[...] + WHISK:	61.4%	64.4%	62.9%	46.0%
submitted:	60.8%	60.8%	60.8%	48.7%
IRISA-TexMex (winner):	48%	72%	57%	46%

Table 6: Performances of final sets of rules on test data

The subtask 1 of the BB BioNLP-ST ranks competitors using the SER measure that must be as close as possible to 0. We are quite close to the winner with a SER of 48.7% against 46%. Our F-measure (60.8%) is even better than the winner's F-measure (57%). Without our mistake, we would have been placed equal first with a far better F-measure (62.9%). We can also notice that the WHISK rule set contribution is negative while it was not the case on the developement data.

8 Conclusion and perspectives

Given the wealth of the OntoBiotope ontology provided for subtask 1 of the BB BioNLP-ST, we have decided to use a method that consists in identifying Bacteria Habitats using information available in this ontology. The method we have used is rule-based and allows the automatic establishment of a set of rules, written in the TextMarker language, that match every ontology element (Habitat Category) with its exact name, exact synonyms or related synonyms in the text. As expected, this method has achieved good results improved by adding a rote learning technique based on training examples and filtering techniques that eliminate categories that don't perform well on the development set.

The WHISK algorithm was also used to learn Bacteria Habitats Categories. It gives a good precision but a low recall because of the poverty of training data. Its combination with the ontology projection method improves the recall and F-measure in developement data but not in the final test data.

The combination of these sources of rules leads to good results with a SER measure close to the winner and a best F-measure.

Actually, due to implementation limitations, WHISK rules are essentially based on the Token level (inflected form) of the corpus. Improvements can be made by ameliorating this implementation

considering the lemmatized form of words, their postags and also terms extracted by a term extractor. There is also another way of improvement that consists in taking into account the *is a* relation of the ontology, both on WHISK rule set and on ontology-based projection rules. Last, a closer look at false positive and false negative errors can lead to some improvements.

Acknowledgments

This work was realized as part of the Quaero Programme funded by OSEO, French State agency for innovation.

References

Mary Elaine Califf and Raymond J. Mooney. 2003. Bottom-up relational learning of pattern matching rules for information extraction. *J. Mach. Learn. Res.*, 4:177–210, December.

Fabio Ciravegna. 2000. Learning to tag for information extraction from text. In *Proceedings of the ECAI-2000 Workshop on Machine Learning for Information Extraction*.

Fabio Ciravegna. 2001. (lp)2, an adaptive algorithm for information extraction from web-related texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*.

Fabio Ciravegna. 2003. (lp)2: Rule induction for information extraction using linguistic constraints. Technical report.

Hamish Cunningham, Diana Maynard and Valentin Tablan. 2000. JAPE: a Java Annotation Patterns Engine (Second Edition). Technical report, of Sheffield, Department of Computer Science.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Kevin S. Mccurley, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. 2003. A case for automated large scale semantic annotations. *Journal of Web Semantics*, 1:115–132.

David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10:327–348.

Dayne Freitag and Nicholas Kushmerick. 2000. Boosted wrapper induction. pages 577–583. AAAI Press.

Wiktoria Golik, Robert Bossy, Zorana Ratkovic, and Nédellec Claire. 2013. Improving Term Extraction with Linguistic Analysis in the Biomedical Domain. *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing13), Special Issue of the journal Research in Computing Science*, pages 24–30.

Peter Kluegl, Martin Atzmueller, Tobias Hermann, and Frank Puppe. 2009. A framework for semi-automatic development of rule-based information extraction applications. In *Proc. LWA 2009 (KDML - Special Track on Knowledge Discovery and Machine Learning)*, pages 56–59.

Nicholas Kushmerick, Daniel S. Weld and Robert Doorenbos. 1997. Wrapper induction for information extraction. In *Proc. Int. Joint Conf. Artificial Intelligence*.

Haibin Liu, Tom Christiansen, William A. Baumgartner, and Karin Verspoor. 2012. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of biomedical semantics*, 3(1):3+.

Stephen Soderland, Claire Cardie, and Raymond Mooney. 1999. Learning information extraction rules for semi-structured and free text. In *Machine Learning*, pages 233–272.

Lawrence H. Smith, Thomas C. Rindflesch and W. John Wilbur. 2004. MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics (Oxford, England)*, 20(14):2320–2321, September.